Technische Universität Berlin **Summer Term 2014**
Fakultät II - Mathematik und Naturwissenschaften
Institut für Mathematik
Andreas Steinbrecher, Jeroen Stolwijk

Numerical Analysis II

Homework Sheet 5

Exercises

Tutorial on May 19

1. **Problem**
Compare the stability regions of Heun's method and the implicit trapezoidal rule. Which method is better for stiff problems?

2. **Problem**
Give an overview of the idea behind extrapolation methods.

3. **Problem**
Compute the method that extrapolates the forward Euler method with the sequence $(n_1, n_2) = (1, 3)$ using the Neville-Aitken scheme. Is this a Runge-Kutta-method? What is its order of consistency?

Theoretical Homework

Due: May 27, during the lecture

1. **Problem** (8 Points)
Let $f(t, y)$ be sufficiently smooth. Consider the value $T_{3,3}$ from the lecture, that is obtained by extrapolating the forward Euler method using the sequence $(n_i)_{i=1}^3 = (1, 2, 3)$. What method does $T_{3,3}$ correspond to? Is it a Runge-Kutta method? If so, give its order, its stage, and its Butcher tableau.

2. **Problem** (5 Points)
Compute the stability function of the Hammer & Hollingsworth method

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 2/3 & 1/3 & 1/3 \\ \hline & 1/4 & 3/4 \end{array}$$

and plot its stability region with Matlab. Is this method A, A(α), or A(0)-stable?

3. **Problem** (7 Points)
Consider the method that is received by taking two steps of Heun's method using the step size $h/2$ and then applying a Richardson extrapolation. Compute the stability function of that method and plot the stability region using Matlab. Is it a Runge-Kutta method?

Total Points: 20

Programming Homework

Due: May 22 (first chance) or June 2 (second chance)

Attention: You can only submit your program on June 2 if you presented a programming approach on May 22!

Write a program that solves an ordinary differential equation $\dot{y}(t) = f(t, y(t))$, $y(t_0) = y_0$, on an interval $[t_0, t_0 + a]$ using an extrapolation method. Your function should be called with the line

$$[\mathtt{h}, \mathtt{t}, \mathtt{u}] = \mathtt{extrapol}(\mathtt{fun}, \mathtt{t0}, \mathtt{y0}, \mathtt{N}, \mathtt{a}, \mathtt{p}, \mathtt{k}).$$

Here, `fun` should be a MATLAB function handle corresponding to the right hand side $f(t, y)$ of the differential equation. It should also be possible for `y` and `f` to be vectors of $\mathbb{R}^n$. The parameter $\mathtt{t0} = t_0$ is the lower interval bound, $\mathtt{y0} = y_0 \in \mathbb{R}^n$ is the initial value, `N` is the number of steps, and $\mathtt{a} = a$ is the interval length. Further, `p` is the order of consistency of the used basic method and `k` is the number of extrapolation stages.
The routine should return the constant step size $\mathtt{h} = \mathtt{a}/\mathtt{N}$, the vector of grid points $\mathtt{t} = [t_0, t_1, \ldots, t_N]$, and the corresponding approximated solution $\mathtt{u} = [u_0, u_1, \ldots, u_N]$.
It should be possible to choose the parameter `p` from $1, 2, 3, 4$ where you get the basic methods $\mathtt{p} = 1$ − forward Euler, $\mathtt{p} = 2$ − Heun's method, $\mathtt{p} = 3$ − Simpson's rule (see homework sheet 4), and $\mathtt{p} = 4$ − classical RK4 (see the document ExpRKMethods.pdf on the website). For each step performed using the basic method, compute $k - 1$ additional approximations of u_{m+1} using the same basic method but n_i steps with the step size h/n_i, where $n_i = 1, 2, 5, 10, 20, 40, 80, 160, 320, 640, \ldots$. Having these k approximations of u_{m+1} for the step sizes $h/n_1, \ldots, h/n_k$, your program should then extrapolate these values of u_{m+1} to the step size $h = 0$. Hence, your program should include the special case of no extrapolation for $\mathtt{k} = 1$.
The extrapolation itself should be performed by using the Neville-Aitken scheme for $\mathtt{p} = 1$ and the linear system involving the Vandermode-like matrix from the lecture otherwise.
To test your program, use the initial value problems

1. $\dot{y}(t) = 2y(t) - e^t, \quad y(0) = 2, \quad t \in [0, 1],$
   with exact solution $y(t) = e^t + e^{2t}$,

2. $\dot{y}(t) = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} y(t), \quad y(0) = [1\ 0]^T, \quad t \in [0, 1]$,
   with exact solution $y(t) = \frac{1}{2} \begin{bmatrix} 1 \\ -1 \end{bmatrix} e^{3t} + \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} e^t$,

3. $\dot{y}(t) = -\tan(t) y(t), \quad y(0) = 1, \quad t \in [0, 6],$
   with exact solution $y(t) = \cos(t)$.

Run your methods on each test problem and compare your approximations to the exact solutions using $N = 5, 10, 20, 50, 100$, and k being less than or equal to 10. Compare the lower order methods with more extrapolation stages to the higher order methods with fewer extrapolation stages. From log-log plots, what can you infer about the order of consistency of an extrapolation method with k extrapolation stages and a p-th order consistent basic method?